SVEUČILIŠTE U ZAGREBU

PRIRODOSLOVNO-MATEMATIČKI FAKULTET

BIOLOŠKI ODSJEK

ODABRANE METODE STROJNOG UČENJA I NJIHOVA PRIMJENA U MOLEKULARNOJ BIOLOGIJI

CHOSEN MACHINE LEARNING METHODS AND THEIR APPLICATION IN MOLECULAR BIOLOGY

SEMINARSKI RAD

Mislav Acman

Preddiplomski studij molekularne biologije

(Undergraduate Study of Molecular Biology)

Mentor: prof. dr. sc. Kristian Vlahoviček

Zagreb, 2015.

SADRŽAJ

1. Uvod

1.1. Strojno učenje

Količina podataka pohranjena u svjetskim bazama podataka udvostručuje se svakih dvadeset mjeseci, a od ukupne količine podataka u svijetu, devedeset posto podataka je nastalo u protekle dvije godine. (Cleophas i sur. 2014)

Trend eksponencijalnog rasta količine podataka prisutan je i na području molekularne biologije, odnosno bioinformatike. Možda najbolji primjer za to je sve veći broj sekvenciranih genoma koji se javljaju zbog sve jeftinijih i pouzdanijih tehnologija za sekvenciranje genoma. Pojavom strojeva za sekvenciranje nove generacije (NGS, od eng. Next-Generation Sequencing), kao što su Illumina i Roche 454, broj čitanja po genomu povećao se za 100 puta, dok se procesiranje po stroju povećalo za 500,000 puta. (Baker 2010)

S velikim količinama dolazi i sve veća kompleksnost podataka, pa tradicionalne statističke metode teško razlučuju iznimke, pravilnosti i obrasce među podacima. Ipak, novije metode obuhvaćene terminom strojnog učenja, omogućuju zaobilaženje tih ograničenja. (Cleophas i sur. 2014)

Pored molekularne biologije, strojno učenje ima primjenu u mnogim drugim područjima kao što su: umjetna inteligencija, procesiranje jezika, pretraživači, medicinska dijagnoza, analiza tržišta dionica, računalne igrice i drugo. (Zhang i sur. 2009)

Strojno učenje je područje računalnih znanosti koje kroz razvoj algoritama omogućuje računalima da nauče, modeliraju i razumiju kompleksne skupove podataka. Algoritam strojnog učenja je računalni proces koji koristi unesene podatke kako bi izvršio određeni zadatak. Strojevi programirani algoritmom postaju sve bolji u obavljanju određenih zadataka što imaju više iskustva i ponavljanja procesa na podacima. Taj proces učenja na unesenim podacima naziva se trening (od eng. training). Drugim riječima, algoritam se optimizira na unesenim podacima za trening kako bi pružio željene rezultate, a procedura se zatim generalizira na nove, prethodno

ne korištene podatke. (Naqa i sur. 2015)

Između algoritama koji se koriste za strojno učenje postoje dvije temeljne podjele:

1. Algoritmi s nadzorom, te algoritmi bez nadzora (od eng. *Supervised and Unsupervised Learning*)

2. Regresijski i klasifikacijski algoritmi (od eng. *Regression and Classification Algorithms*)

1.2. Algoritmi s nadzorom i bez nadzora

Algoritmi s nadzorom pokušavaju povezati dva tipa podataka: prediktore i odgovor (od eng. *predictor and response*). Ako prediktore predstavimo s varijablom X ($X = X_1, X_2, ..., X_n$), a podatke odgovora s varijablom Y ($Y = Y_1, Y_2, ..., Y_n$), onda će algoritmi s nadzorom nastojati povezati ta dva tipa podataka pomoću određene funkcije f od X tako da vrijedi:

$$Y = f(X) + \varepsilon \qquad [1]$$

za sve prediktore i odgovore.

Iz gornje formule [1] vidimo kako procjenu varijable Y prati slučajna pogreška ε. Stoga, procijenjene funkcije f gotovo nikad neće apsolutno točno odrediti odgovor Y zbog slučajne pogreške u samim podacima na kojima je algoritam treniran.

Proces traženja najbolje funkcije f koja opisuje odnos između prediktora i odgovora nazivamo uklapanje modela. Algoritmi s nadzorom nastoje uklopiti model koji će što preciznije predvidjeti odgovor za novo unesene podatke ili koji će pomoći u boljem razumijevanju odnosa između odgovora i prediktora.

Druga vrsta algoritama kao konačan cilj nastoji opisati odnos između podataka varijable X. Takvim algoritmima nije smisao predviđanje odgovora Y niti procjena funkcije f, već nastoje razlučiti odnose između unesenih podataka X. Ovakav tip algoritma naziva se algoritam bez nadzora jer ne postoji varijabla Y koja bi nadzirala analizu algoritma, zato su takvi algoritmi često složenij i izazovniji, jer ne možemo

provjeriti rezultat algoritma. Većina algoritama bez nadzora kao konačan cilj ima grupiranje podataka (od eng. *cluster analysis*) s obzirom na neke prepoznate karakteristike unesenih podataka.

Važnost strojnog učenja bez nadzora raste u mnogim područjima znanosti, gospodarstva, industrije, medicine i dr. Jedan od primjera koji to potvrđuje je tipizacija tumora, gdje pacjente možemo svrstati u određenu skupinu s obzitom na ekspresiju gena u tumoru kako bismo bolje razumjeli bolest.

1.3. Regresijski i klasifikacijski algoritmi

Svojstva koja se koriste u analizi strojnim učenjem razvrstavaju se u dvije skupine: kvantitativna svojstva i kvalitativna svojstva. Kvantitativnim svojstvima pripisuju se različite numeričke vrijednosti u određenoj kategoriji (npr. visina, dob, prinos, energija itd.), dok kvalitativna svojstva vrijednujemo prema pripadnosti određenom razredu ili kategoriji (npr. spol, krvna grupa, vrsta materijala, boja itd.).

Algoritmi kojima se razmatraju kvantitativna svojstva nazivaju se regresijski algoritmi, a oni koji proučavaju kvalitativna nazivaju se klasifikacijski.

Međutim, svrstavanje algoritama u jednu od te dvije skupine nije uvijek moguće zato što se mnogi algoritmi strojnog učenja mogu nositi s obje vrste problema, klasifikacijom i regresijom. Neki od primjera takvih algoritama su “najbliži K-ti susjed” (od eng. *K-nearest neigbors*) i “Random Forests”.

2. Random Forests

2.1. Osnove izgradnje stabla odluke

Random forests jedna je od metoda strojnog učenja koja se temelji na izgradnji stabla odluke (od eng. *decision tree*). Izgradnja stabla podrazumijeva segmentiranje prediktorskog prostora (prediktorskih vrijednosti za trening) u jednostavnije regije pomoću niza pravila dijeljenja. Algoritam kasnije predviđa vrijednosti odgovra s obzirom na novo uneseni prediktor, odnosno opservaciju, vodeći se pravilima za

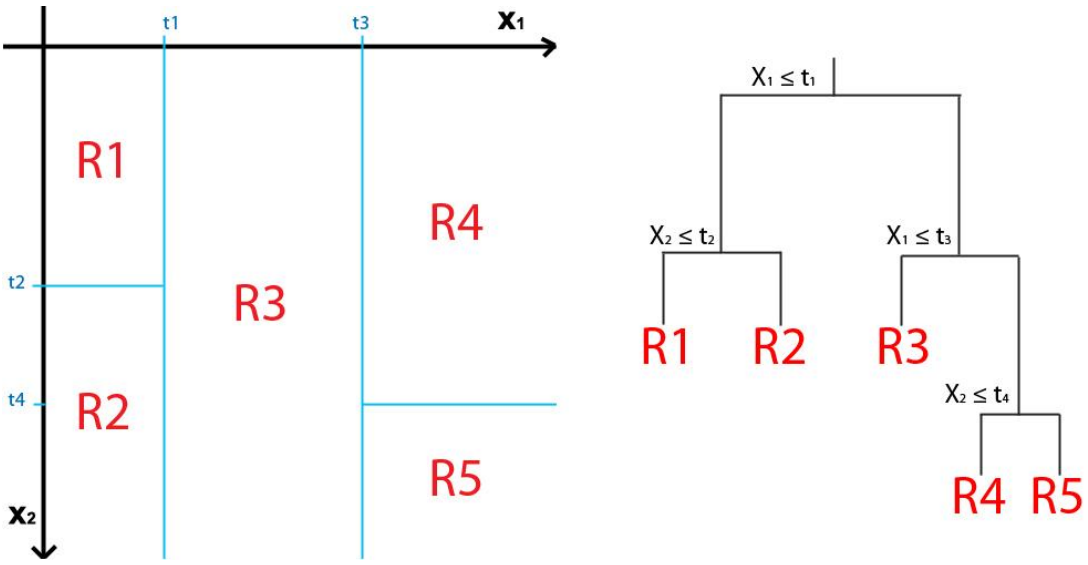
dijeljenje. Jasno je kako su metode temeljene na izgradnji stabla zapravo algoritmi s nadzorom, a mogu se koristiti za rješavanje klasifikacijskih i regresijskih problema.

Za rješavanje regresijskih problema izrađuju se regresijska stabla u dva osnovna koraka:

1. Dijeljenje prediktorskog prostora: moguće vrijednosti prediktora X1, X2, ..., Xp podijele se u J različitih nepreklapajućih regija R1, R2, ..., RJ (slika 1.)

2. Svakoj novoj opservaciji koja pripada nekoj od regija R1, R2, ..., RJ pripišemo vrijednost odgovora te regije. Kod regresijskih stabala to je najčešće srednja vrijednost odgovora korištenih za trening.

a) b)



Slika 1. a) Grafički prikaz podjele prediktorskog prostora za jednostavno regresijsko stablo. b) Shematski prikaz jednostavnog regresijskog stabla s pravilima dijeljenja za slučaj pod a).

Za dijeljenje prediktora u regije najčešće se koristi metoda rekurzivnog binarnog cijepanja. Metoda kreće od vrha stabla (kad sva opažanja pripadaju jedinstvenoj regiji), te u svakom koraku binarno cijepa prediktorski prostor. Svako cijepanje stvara dvije nove grane, odnosno regije, a u svakom idućem koraku odabire se jedna od regija koja će biti pocijepana. Metoda je sebična zato što u svakom koraku izgradnje stabla odabire najbolje moguće cijepanje za taj korak. Savršeno stablo

odluke nemoguće je dobiti jer bi u tom slučaju metoda morala gledati unaprijed, tj. odabirati cijepanja koja bi dovela do boljeg stabla u nekom budućem korak. Takav pristup drastično bi usporio proces izgradnje stabla, pa bi algoritam postao neefikasan.

Klasifikacijsko stablo koristi se za predviđanje kvalitativnog odgovora, a princip izgradnje sličan je regresijskom stablu. Klasifikacijskim stablom ne predviđa se srednja vrijednost odgovora neke regije, nego se kao predviđeni odgovor uzima najučestalija kategorija određene regije prisutna kod podataka za trening.

Za izgradnju klasifikacijskog stabla također se koristi metoda rekurzivnog binarnog cijepanja. Prilikom odabira najboljeg mjesta za cijepanje, odnosno za stvaranje novih regija, u obzir se uzima "čistoća" određene grane. Drugim riječima, što je više odgovora u određenoj grani iste kategorije to je odabrano cijepanje povoljnije za ukupnu preciznos stabla. Za procjenu kvalitete grananja najčešće se koriste tri pristupa: klasifikacijski stupanj pogreške (od eng. *classification error rate*), Gini indeks i unakrsna entropija.

2.2. Bootstrap metoda

Bootstrap je široko primjenjiva i izrazito moćna statistička metoda koja se koristi u random forests algoritmu kako bi se smanjila varijanca stabala i tako povećala preciznost predviđanja. Naime, ako se podatci za trening nasumično podijele u dvije skupine i pomoću svake polovice izgradimo bilo regresijska ili klasifikacijska stabla, predviđani rezultati mogu se jako razlikovati između stabala.

Ideja bootstrapa je da se varijanca među podacima smanjuje usrednjavanjem. Kod algoritma random forests to se postiže tako da se iz podataka za trening nasumično uzimamaju uzorci. Iz dobivenih nasumičnih uzoraka algoritam izgradi "šumu" stabala (za svaki uzorak po jedno stablo).

Kod velikog broja stabala ne može se dobiti zoran prikaz konačnog procesa strojnog učenja. Konačan proces učenja nije moguće prikazati samo jednim stablom zato što se stabla međusobno razlikuju. Nemoguće je odrediti koje su grane najvažnije za proces, tj. koje bi se trebale nalaziti u shemi procesa učenja. Stoga će u slučaju regresijskog stabla konačn odgovor biti prosjek odgovora "šume" stabala, a kod

predviđanja kvalitativnog odgovora (klasifikacijska stabla) kao istinu uzimamo odgovor većine stabala. Random forests metoda povećava preciznost odgovora nauštrb interpretativnosti.

2.3. Još jedna razina nasumičnosti

Pretpostavimo da podaci za trening sadrže jedan snažan prediktor i nekoliko srednje jakih prediktora. Prilikom gradnje nasumične "šume" većina stabala će koristiti snažan prediktor za grananje u vrhu stabla. Posljedica toga je da većina stabala u "šumi" nalikuje jedno na drugo. Kažemo da su stabla visoko korelirana. Usrednjavanje visoko koreliranih stabla ne dovodi do značajnog smanjenja varijance, te se tako gubi na preciznosti predviđanja.

Random forests algoritam će prilikom gradnje stabla ograničiti izbor prediktora za svako grananje. Ako je p ukupan broj prediktora u podacima za trening koji se mogu razmotriti, algoritam će prilikom biranja novog prediktora za grananje izbor suziti na $\sqrt{p}$ prediktora. Za svako grananje uzima se novi slučajni uzorak prediktora na razmatranje.

Random forests uvjetuje odabir iz podskupa prediktora i tako slabijim prediktorima daje priliku da više utječu na odgovor, smanjuje varijancu između stabala, ubrzava rad algoritma i povećava preciznost. (Breiman 2001)

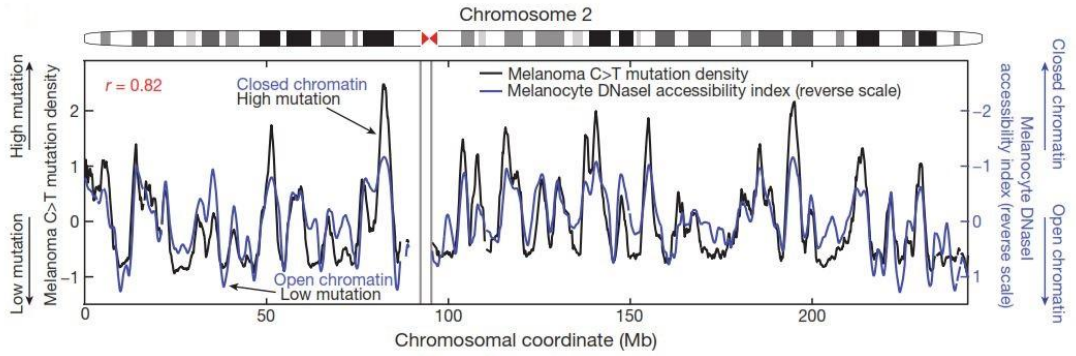
2.4. Primjena

Primjena algoritma random forests zaista je široka. Algoritam je vrlo elastičan i primjenjiv na različitim vrstama bioloških podataka: fenotipske karakteristike, genske sekvence, epigenetske karakteristike, proteinske parametre i razna druga mjerenja.

Jedan od primjera primjene random forests algoritma na kvalitativnim podacima predstavljen je u radu od Polak, Karlić i sur. u veljači 2015. Istraživanje je obuhvatilo 173 genoma iz osam različitih vrsta raka: melanom, plućni adenokarcinom,

rak jetre, kolorektalni rak, glioblastom itd. Genomska distribucija mutacija u svakom od tih genoma uspoređena je s podacima o 424 epigenetičke značajke za 106 različitih netumorskih tipova stanica izmjerenim od strane konzorcija Epigenome Roadmap.

Naime, uspoređujući stanice raka i zdrave ishodišne stanice primjećeno je kako su epigenetske značjke eukromatina povezane s niskom mutacijskom gustoćom, dok su epigenetičke značajke heterokromatina povezane s visokom mutacijskom gustoćom. Na slici 2. vidljivo je kako distribucija epgenetskih značajka u genomu ishodišnih stanica (melanocita) korelira s mutacijskom gustoćom stanica melanoma.



Slika 2. Gustoća C>T mutacija u stanicama melanoma odgovara kromatinskim karakteristikama melanocita. Crna linija označava mutacijsku gustoću. Plava linija označava indeks pristupačnosti DNase I melanocitnom kromatinu u okviru od sto tisuća parova baza. Više vrijednosti odgovaraju manje pristupačnom kromatinu odnosno većoj gustoći mutacija. (preuzeto iz Polak i sur. 2015)

Kako bi povezali različite kromatinske značajke s mutacijskom gustoćom, a tako i s vrstom ishodišne stanice korišten je random forests algoritam. Algoritam je pomoću šume od 1000 stabala radio uspješna predviđanja prilikom testiranja na sve mutacije ili samo na predominantnu vrstu mutacije. Pokazano je kako su za preciznost predviđanja najviše zaslužne značajke kromatina, dok su genska ekspresija i sadržaj nukleotida bili preslabi prediktori: preciznost algoritma s genskom ekspresijom kao prediktorom bila je 78% i 57% u dva pokušaja.

Rezultati dobiveni u istraživanju pokazali su da se samo pomoću obrasca mutacija može predvidjeti ishodišna vrsta stanica za tumor neke osobe. Takav pristup

je s 88% uspješnosti povezao testirane vrsta raka sa ishodišnim stanicama. Razvijenom metodom tima Polak, Karlić i sur. i sekvenciranje genoma tumora s nepoznatim ishodištem, moguće je vrlo precizno identificirati ili okarakterizirati ishodišnu vrstu stanice. (Polak i sur. 2015)

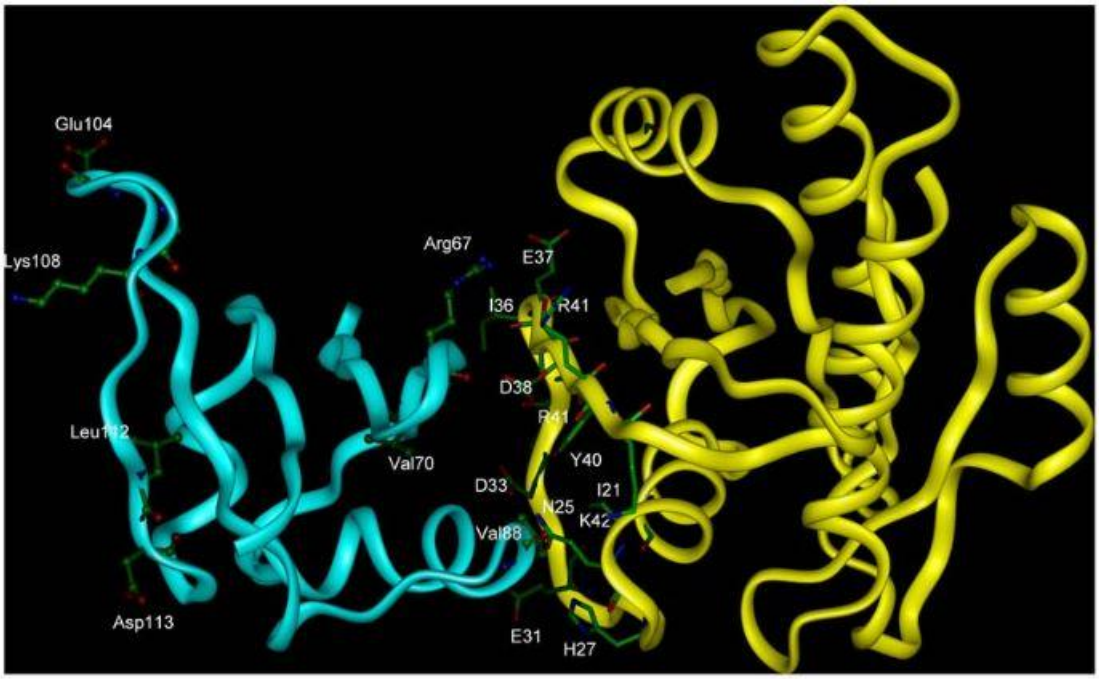
Drugačiji primjer implementacije random forests algoritma možemo pronaći u radu iz 2009. godine tima znanstvenika Šikić, Tomić i Vlahoviček. Oni su pomoću random forests algoritma doprinijeli su rješavanju jedanog od velikih izazova molekularne biologije, biokemije, sistemske biologije i razvoja lijekova: predviđanje mjesta protein-protein interakcije. U radu su predstavljene dvije metode za predviđanje interakcija od kojih se jedna temelji samo na informacijama proteinske sekvence, a druga na kombinaciji proteinske sekvence i karakteristika trodimenzionalne strukture.

Za trening i testiranje metoda korišteni su podaci 1134 proteinska lanaca i 333 kompleksa. Iz tog skupa podataka svi aminokiselinski parovi koji su se nalazili unutar 6 Å označeni su kao interagirajući, dok se za preostale smatralo da ne postoji interakcija. Prediktori za predviđanje interagirajućih aminokiselina konstruirani su u obliku prozora od devet aminokiselina u nizu. Razred u koji je svrstavan pojedini prozor određen je s obzirom na broj interagirajućih aminokiselina unutar prozora. Tako organizirani podaci analizirani su random forests algoritmom.

Prilikom razvoja druge metode, među prediktore uvršteni su i podaci o trodimenzionalnoj strukturi proteina. Među svim dostupnim informacijama o trodimenzionalnoj strukturi odabrana su samo ona svojstva koja najbolje predviđaju iteragirajuća mjesta. Najjači među prediktorima bila je naravno primarna struktura proteina, a iza nje su slijedili: nepolarna i relativno nepolarna pristupačnost površine (od eng. *non-polar accessible surface area*), indeks maksimalne dubine (od eng. *maximum depth index*), prosječan indeks dubine (od eng. *average depth index*), te indeks minimalnog izbočenja (od eng. *minimum protrusion index*).

Autori su svoju metodu testirali u predviđanju interakcije Ras vezujuće

domene (RBD, od eng. Ras Binding Domain) C-Raf1 proteina i Ras proteina. (slika 3.) Iako su poznate trodimenzionalne strukture Ras i C-Raf1 proteina, struktura kompleksa nije eksperimentalno utvrđena. No, rezultati su pokazali kako se većina predviđenih aminokiselina u interakciji podudara s dosadašnjim eksperimentalinm rezultatima.



Slika 3. Model Raf(plavo)-Ras(žuto) kompleksa s predviđenim interagirajućim aminokiselinama. (preuzeto iz Šikić i sur. 2009)

Razvijena metoda temeljena samo na proteinskoj sekvenci ima preciznost od 84% i uspješno predviđa 26% svih mjesta interakcije (F-mjera je 40%). U kombinaciji sa strukturnom informacijom proteina, preciznost predviđanja je 76% i obuhvaća 38% svih mjesta proteinske interakcije (F-mjera je 51%).

Razni radovi drugih autora koristili su različita trodimezionalna svojstva i algoritme strojnog učenja kako bi razvili pouzdanu metodu za predviđanje protein-protein interakcija. Neki od korištenih klasifikacijkih algoritma su: funkcije bodovanja, metoda potpornih vektora (SVM, od eng. *support vector machines*) i neuralne mreže. Dio tih radova imao je sličnu ili malo veću pouzdanost i preciznost

od predstavljene metode. Međutim, primarni cilj rada Šikića, Tomića i Vlahovičeka bio je poboljšati predviđanje mjesta interakcija u proteinskim kompleksima temeljeno samo na proteinskim sekvencama. (Šikić 2009.)

3. Metoda potpornih vektora

3.1. Razdvajanja podataka hiperravninom

Metoda potpornih vektora (SVM, od eng. *support vector machines*) smatra se jednim od najboljih kreativnih pristup za klasifikaciju podataka. Razvijen je u području računalnih znanosti tijekom 1990tih godina na ideji jednostavnog pristupa: klasifikator maksimalnom marginom (MMC od eng. *Maximal Margin Classifier*).

MMC je način odvajanja dvije klasifikacijske skupine pomoću optimalne (ili maksimalne) razdvajajuće hiperravnine (od eng. *optimal separating hyperplane*). U p-dimenzionalnom prostoru hiperravnina (od eng. *hyperplane*) je naziv za podprostor s dimenzijom p-1. Na primjer, u dvodimenzionalnom prostoru hiperravnina će predstavljati jednodimenzionalan podprostor - pravac.

Hiperravninu u prostoru s p dimenzija matematički definiramo kao:

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_p X_p = 0 \qquad [2]$$

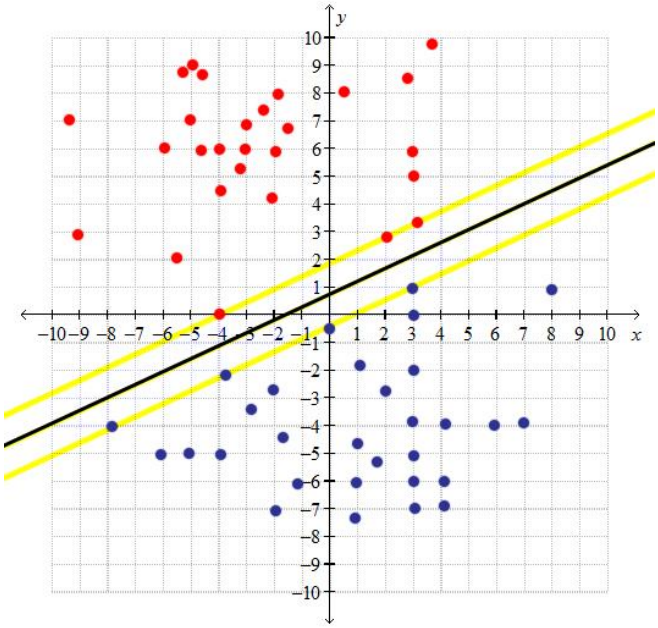
za parametre β_0, β_1, β_2, ... i β_p.

Ukoliko neka opservacija X sa svim svojim pripadajućim svojstvima $X_1,...,X_p$ u p-dimenzionalnom prostoru zadovoljava jednadžbu [2] tada ona leži na hiperravnini. U suprotnom, rješenje jednadžbe za neku opservaciju X može biti veće ili manje od 0, te s obzirom na rješenje, točka je svrstana u jedan od dva razreda koji su nastali dijeljenjem hiperravninom. Budući da se u ovom slučaju hiperravnina koristi za klasifikaciju podataka, često se naziva klasifikatorom.

Marginom se naziva prostor između hiperravnini najbližih točaka i same

hiperravnine. Jasno je da postoji beskonačno mnogo hiperravnina kojima možemo razdvojiti podatke, a svaka od tih hiperravnina ima različitu širinu margine. Prilikom traženja optimalne razdavjajuće hiperravnine odabire se ona koja ja najudaljenija i nalazi se između dva razreda podataka - ima maksimalnu marginu.

Prema opservacijama *X* u p-dimenzionalnom prostoru općenito se odnosi kao prema vektorima. Potporni vektori (od eng. s*upport vectors*) su opservacije koje diktiraju položaj optimalne razdvajajuće hiperravnine. Oni predstavljaju opservacije koje leže na rubovima margine i jednako su udaljene od razdvajajuće hiperravnine. (slika 4.) Promijeni li se položaj nekog od vektora podrške promijenit će se i položaj optimalne razdvajajuće hiperravnine. No, ako se promjeni položaj bilo koje druge točke, to neće utjecati na razdvajajuću hiperravninu.



Slika 4. Dva razreda opservacija razdvojeni optimalnom razdvajajućom hiperravninom. Uzduž hiperravnine označeno je područje margine, a na rubu hiperravnine nalaze se potporni vektori.

Stvarni biološki podaci grafički prikazani često izgledaju neuredno i razredi među podacima nisu jasno definirani. U tom slučaju gornjom metodom nemoguće je pronaći optimalnu razdvajajuću hiperravninu jer bi se neki od podataka mogao naći unutar prostora margina ili čak na pogrešnoj strani hiperravnine.

Problem klasifikacije takvih podataka riješen je propusnom hiperravninom. Drugim riječima, algoritam će točno razdvojiti maksimalan broj podataka u skupine, a samo nekim točkama dopušta ne poštivanje pravila razdvajanja.

Prilikom traženja optimalne razdvajajuće hiperravnine definira se vrijednost C koja će omogućiti praćenje takvih pogrešaka. U konačnici odabrana hiperravnina ne smije imati više od C pogrešaka na podacima za trening. Vrijednost parametra C ručno se određuje i tako regulira osjetljivost algoritma, a posredno regulira i širinu margine, te sam položaj hiperravnine. Na primjer, ako je C vrijednost velika algoritam će dopuštati puno pogrešaka te će moći povećati prostor margine, a ako je C mali morat će tražiti uske margine koje će podatci rijetko prelaziti.

Pored opservacija koje leže na margini, veliki utjecaj na položaj optimalne hiperravnine imaju i podaci koji ne poštuju pravila razdvajanja (pogreške), te se oni zajedno nazivaju potpornim vektorima. Ova metoda klasifikacije otporna je na ponašanje opservacija koje se nalaze daleko od hiperravnine jer je ona pod utjecajem malog podskupa podataka za trening.

3.2. Klasifikacija kernelima

Metoda za klasifikaciju propusnom hiperravninom je jednostavan i intuitivan pristup za razvrstavanje podataka u podgrupe ako je granica među podgrupama linearna. No, samo traženje optimalne razdvajajuće hiperravnine je nešto kompleksnije. Budući da opservacije promatramo kao vektore, prilikom traženja optimalne razdvajajuće hiperravnine sama procjena parametara, a time i oblika jednadžbe, te klasifikacija svake nove opservacije koja se unosi u algoritam, svodi se na računanje unutarnjeg produkta vektora podrške. Funkcija optimalne radvajajuće hiperavnine, tj. klasifikatora piše se kao:

$$f(x) = \beta_0 + \sum_{i \in S} \alpha_i \langle x, x_i \rangle ,$$

[3]

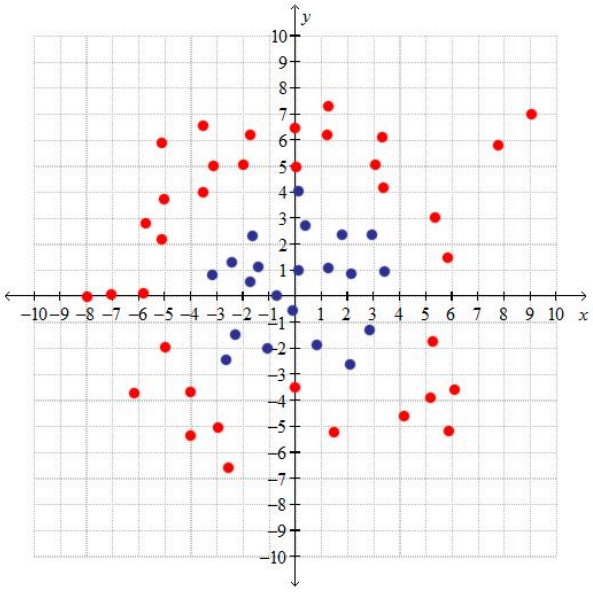
gdje su β_0 i α_i parametri funkcije, S je skup svih vektora podrške, a $\langle x, x_i \rangle$ je unutarnji

produkt x i vektora podrške x_i.

U slučajevima razdvajanja podataka s nelinearnim granicama koristi se pristup kernela. Osnovna ideja je da se svi unutarnji produkti koji se pojavljuju prilikom računanja zamjene s funkcijom K koja se naziva kernel. Funkcija K je generalizacija unutarnjeg produkta koja će, kao i unutarnji produkt, računati sličnost dvije opservacije.

Postoji mnogo vrsta kernela od kojih su najpopularniji: linearni kernel (koji se koristi za računanje razdvajajuće hiperravnine), polinomalni kernel i radijalni kernel.

Radi jasnije predodžbe ideje kernela, razmatraju se podaci sa slike 5. Podaci ne mogu biti razdvojeni pravcem. Međutim, svakom podatku može se izračunati udaljenost od ishodišta i razdvojiti ih optimalnom razdvajajućom hiperravinom na temelju te udaljenosti. No, u algoritmu SVM nije potrebno transformirati podatke i računati sve udaljenosti, jer su svi podaci u funkciji razdvajajuće hiperravnine predstavljeni unutarnjim produktom. Ako predstavimo unutarnji produkt podataka na drugačiji način, možemo uspješno razdvojiti kompleksne skupove podataka. (Herbich i sur. 2012)



Slika 5. Skup podataka koje nije moguće razdvojiti optimalnom razdvajajućom hiperravninom.

Velika prednost SVM je mogućnost rada s velikom količinom podataka. Također, metoda je otporna na ekstremne vrijednosti podataka dok god poštuju pravila razdiobe. Iako je u ovom pregledu predstavljena razdioba opservacija u dvije skupine, SVM se može koristiti i za klasifikaciju u više razreda. Predviđanje u više razreda istovremeno je zahtjevniji postupak jer klasifikacijski algoritam mora naučiti razlikovati i konstruirati veći broj granica i pravila za razdiobu.

3.3. Primjena u klasifikaciji tumora

SVM algoritam može se primijeniti za rješavanje različitih problema u molekularnoj biologiji, biokemiji i medicini. Neki od primjera su: selekcija gena odgovornih za pojedine bolesti; predviđanje ishoda liječenja; identifikacija i klasifikacija gena detektiranih RNA-microarrayem ili predviđanje strukture i funkcije proteina. (Naul 2009, Mukherjee 2003)

Za uspješnu klasifikaciju tumora potrebne su nam informacije o ekspresiji gena u tumorskim stanicama koje na jednostavan način možemo prikupiti zahvaljujući microarray tehnologiji.

Jedan od prvih uspješnih pokušaja klasifikacije raka SVM predstavljen je u radu Goluba, Slonima et.al. 1999. Njihov algoritam treniran je na 38 uzoraka analiziranih DNA microarrayom od kojih je 11 pripadalo akutnoj mijeloidnoj leukemiji, a 27 akutnoj limfatičnoj leukemiji. Pouzdanost treniranog klasifikatora mjerena je na 35 testnih uzoraka. SVM algoritam s linearnim kernelom uspješno je razlučio 34 od 35 uzoraka. (Golub i sur. 1999.)

U gornjem primjeru, korištenje polinomalnih i radijalnih kernela nije poboljšalo pouzdanost klasifikatora. Međutim, kada su uklonjeni geni važni za klasifikaciju, korištenje polinomalnog kernela popravilo je učinak algoritma. Vidljivo je kako prilikom klasifikacije tumora informacije o određenim tipovima gena imaju jako važnu ulogu. (Mukherjee 2003)

Klasifikacija tkivnih tumora, za razliku od leukemija, pokazao se težim

izazovom. Problemi se javljaju zbog ograničenja u količini, identifikaciji, pripremi i homogenosti uzoraka. Tkivni tumori su heterogeni po sastavu, pa ekspresijski profili često sadrže udio zdravih stanica.

Unatoč problemima, tijekom godina razvijena je uspješna metoda za opću (višerazrednu) klasifikaciju tumora pomoću SVM samo na temelju informacija o genskoj ekspresiji. Algoritam je treniran na 198 DNA-microarray uzoraka iz 14 najučestalijih tipova tumora. Ukupna preciznost klasifikacije je 78%, a korištenjem većeg broja uzoraka algoritam može postići i do 90% preciznosti. (Rifkin 2003)

Jednostavnost, pristupačnost, brzina i preciznost odrađivanja tipa tumora neupitno je važna za uspješno liječenje ove teške bolesti. Zasigurno će sve veći broj sekvenciranih tumorskih genoma i sve veća količina informacija o ekspresiji gena uvelike pridonijeti uspješnijoj klasifikaciji tumora.

4. Algoritam *K*-srednjih vrijednosti

4.1. Metoda grupiranja

Metode grupiranja pripadaju skupini algoritama za strojno učenje bez nadzora. Njima ne nastojimo predviđati, već otkriti strukturu i zanimljiva opažanja određenog skupa podataka. Razvijene su mnoge tehnike za grupiranje podataka s obzirom na sličnosti i zajednička svojstva. Prilikom izdvajanja podgrupa nastoji se postići homogenost, odnosno da je unutar svake podgrupe što manja varijanca među podacima, dok se izdvojene podgrupe međusoobno što više razlikuju.

Osim u biologiji, metode grupiranja koriste se u mnogim drugim područjima kao što su ekonomija, zdrastvo, marketing, računalni programi itd. Među mnogim metodama za grupiranje koje su razvijene unutar tih područjima, postoje dvije koje se ističu kao najpoznatije i najšire korištene: algoritam *K*-srednjih vrijednosti (od eng. *K*-means clustering) i hijerarhijsko grupiranje (od eng. *hierarchical clustering*).

Hijerarhijsko grupiranje koristi se za pronalaženje svih mogućih smislenih podgrupa među podacima. Rezultat hijerarhijskog grupiranja je dendogram koji prikazuje opservacije i moguće podgrupe tih podataka.

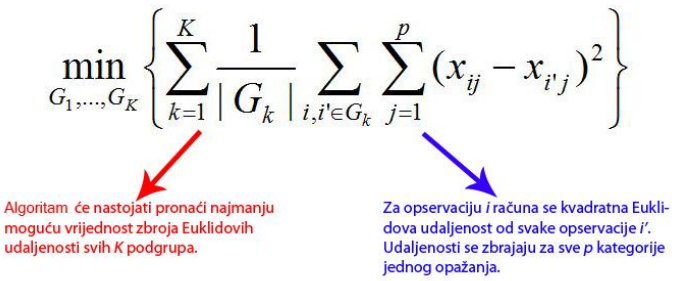
U biologiji se češće koristi algoritam *K*-srednjih vrijednosti, koji za razliku od hijerarhijskog grupiranja nastoji podijeliti podatke u unaprijed određene *K* podgrupe. Sve 1, ..., n opservacije u našem skupu podataka, kao i podgrupe: G_1, G_2, ..., G_K u koje su opservacije podijeljene slijede dva pravila:

1. $G_1 \cup G_2 \cup \ldots \cup G_K = \{1, \ldots, n\}$. Svaka opservacija pripada najmanje jednoj od *K* podgrupa.

2. $G_k \cap G_{k'} = \{/\}$ za sve $k,k' \in \{1, \ldots K\}$ i $k \neq k'$. Podgrupe se ne preklapaju. Svaka opservacija pripada samo jednoj podgrupi.

4.2. Algoritam

Cilj algoritma *K*-srednjih vrijednosti je razvrstati podatke u *K* podgrupa, tako da je unutar svake podgrupe minimalna razlika među opservacijama. Razlika među opservacijama unutar neke podgrupe mjeri se pomoću kvadrirane Euklidove udaljenosti. Kvadriranu Euklidovu udaljenost nastojimo minimizirati za sve *K* podgrupe (slika 6.)



Slika 6. Formula kojom je vođen algoritam *K*-srednjih vrijednosti i objašnjenje formule.

Traženje minimalne kvadrirane Euklidove udaljenosti u nekom skupu od n podataka nije trivijalan problem, budući da postoji K^n načina na koji možemo podijeliti n podataka u K podgrupa. Taj broj je iznimno velik, te je gotovo nemoguće

ispitati sve kombinacije za velike K i n. Kako bi bio što brži i efikasniji, algoritam *K*-srednjih vrijednosti tražit će lokalni optimum. Takav pristup pokazao se zadovoljavajućim rješenjem.

Algoritam djeluje u dva koraka:

1. Svaka opservacija se nasumično svrstava u jednu od *K* podgrupa.
2. Korak 2. se ponavlja sve dok se ne prestanu mjenjati vrijednosti centroida:
    i. Za svaku od podgrupa računa se geometrički centar za sve vrijednosti observacija (centroid).
    ii. Svaka opservacija svrstava se u podgrupu čiji je centroid najbliži vrijednosti te opservacije. Blizina centroida i opservacije računa se pomoću kvadratne Euklidove udaljenosti.

Rezultat algoritma velikim dijelom ovisit će o nasumičnom početnom razvrstavanju opservacija. Kako bi se smanjio utjecaj početnog razvrstavanja i dobio što precizniji rezultat, proces računanja lokalnog optimuma ponavlja se više puta. Od svih dobivenih rezultata odabire se najbolji.

4.3. Primjena u filogenetskoj analizi

Filogenetička analiza, a posebno rekonstrukcija filogenetskog stabla pod utjecajem su dvije vrste pogrešaka: stohastičkih i sistematskih. Dok su stohastičke pogreške neizbježne jer se javljaju zbog rekonstrukcije iz konačnog skupa podataka, sistemske pogreške nastaju zbog primjene neadekvatnog modela molekularne evolucije, te ih je moguće izbjeći. Kako bi se umanjio utjecaj sistemskih pogrešaka važno je unaprijediti pristup odabiru modela molekularne evolucije. Nedavno objavljen rad (Frandsen i sur. 2015) predstavio je novi algoritam nazvan iterativni algoritam *k*-srednjih vrijednosti (od eng. *iterative k-means algorithm*) koji automatizira i poboljšava najčešće korištenu metodu za odabir modela -

particioniranje (od eng. *partitioning*).

Metoda particioniranja odnosi se na cijepanje sravnatih sekvenci u nekoliko kraćih blokova sekvenci. Svakom bloku pridružuje se odgovarajući evolucijski model, te se s tako obrađenim sekvencama kreće u daljnju filogenetsku analizu i rekonstrukciju filogenetskog stabla. Iako postoje elegantnija rješenja kao što su mixture models, prednost particioniranja nad ostalim metodama je primjenjivost prilikom rada s velikim skupom podataka.

Glavni problem particioniranja je odabir sheme za particioniranje, tj. na koliko blokova i kako treba podijeliti sravnate sekvence. Ispitivanje svih mogućih načina particioniranja sravnatih sekvenci nemoguće je izvesti zbog velikog broja mogućih shema. Drugi matematički pristupi nisu primjenjivi na velike skupove podataka ili zahtjevaju od korisnika da unaprijed odrede broj blokova. Tradicionalni način particioniranja podrazumijeva "ručnu" podjelu sravnatih sekvenci prema nekim strukturalnim svojstvima kao što su granice gena, pozicija kodona, struktura rRNA i sl. Ograničenje tradicionalnog particioniranja dolazi do izražaja prilikom analize molekularnih markera koji nisu protein-kodirajuće regije genoma. Novo razvijen algoritam sam će odabrati optimalan broj blokova na koji treba podijeliti sravnate sekvence i jednako je primjenjiv na velike filogenetske skupove podataka i male skupove namjenjene barkodiranju.

Algoritam je temeljen na algoritmu *K*-srednjih vrijednosti i sastoji se od nekoliko koraka:

1. Procjenjivanje početne topologije filogenetskog stabla iz sravnatih sekvenci
2. Odabir najboljeg supstitucijskog modela za početni blok sravnatih sekvenci.
3. Izračun ocjene za trenutnu shemu particioniranja na temelju podudarnosti s današnjim evolucijskim modelom i teorijom.

4. Testiranje svakog bloka u trenutnoj shemi za daljnju podjelu. Svaki blok u trenutnoj shemi podijeli se u dva manja bloka pomoću algoritma *K*-srednjih vrijednosti. Za svaki od dva nova bloka odabire se najbolji supstitucijski model. Ocjenjuje se nova shema particioniranja. Ako je dobivena ocjena bolja od prethodne, blok se označuje za podjelu.

5. Ukoliko nema blokova označenih za podjelu, algoritam se prekida. U suprotnom, svaki označeni blok se podijeli na predviđena dva manja bloka i algoritam nastavlja od koraka 3.

Učinak algoritma testiran je na deset različitih skupova podataka zajedno s nekoliko drugih poznatih pristupa za particioniranje. Iterativni algoritam *k*-srednjih vrijednosti pokazao je superiornost u svih deset slučajeva. (Frandsen i sur. 2015)

5. Zaključak

Predstavljene metode u ovom radu, kao i navedeni primjeri, predstavljaju samo mali dio velikog područja strojnog učenja. To područje neprestano napreduje. Razvijaju se novi i kreativni algoritmi ili se poboljšavaju stari. Velika mogućnost izbora algoritama strojnog učenja pogoduje raznolikim problemima s kojima se susreću bioinformatičari i ostali molekularni biolozi. Na primjer, za analizu velike količine genomskih podataka potreban je algoritam koji ih može brzo i efikasno procesirati, dok je za analizu metagenoma pogodan algoritam bez nadzora.

Iako se neke od navedenih metoda čine kompleksnima, znanstvenicima su dostupni brojni alati koji sadrže razne pakete s implementiranim algoritmima strojnog učenja. Neki od primjera takvih alata je platforma RapidMiner ili programski jezici: R, MATLAB ili Python (s poznatim distribucijama SciPy i NumPy).

Odabir iz spektra dostupnih metoda najvećim dijelom ovisi o vrsti problema s kojim se znanstvenik suočava. Hoće li korišten algoritam biti s nadzorom ili bez

nadzora, regresijski ili klasifikacijski ovisi o skupu podataka koji je potrebno analizirati. Random forests jedna je od najpopularnijih metoda strojnog učenja - brza je i jednostavna. No, ako se skup podataka od interesa sastoji od velikog broja značajki (svojstava), pogodnije je koristiti SVM iako je sporiji i računski zahtjevniji. SVM je moćno oruđe, ali suočeni sa zahtjevnim problemom vjerojatno ćemo posegnuti za danas najmoćnijim algoritmom strojnog učenja - Neuronskom mrežom (od eng. *Neural network*).

Korištenje kompleksnih algoritama nije uvijek isplativo. Neuronska mreža je teško primjenjiva i zahtjevaja veliku količinu podataka kako bi bila uspješna i pouzdana. Ponekad i jednostavniji algoritmi mogu dominirati. Dobar primjer su algoritmi *K*-srednjih vrijednosti i njegov bliži rođak *K*-medoida koji imaju veću fleksibilnost što se tiče veličine skupa podataka.

Iako je odabir metode najbitniji korak u rješavanju bioloških problema i pitanja, ne smiju se izostaviti koraci koji prethode tome. Predprocesiranje skupa podataka i odabir relevantnih svojstava od velike su važnosti za uspješno korištenje odabrane metode. Ti koraci ovise o ljudskoj pogreški, ali i to je moguće zaobići korištenjem Deep Learning seta algoritama koji se pokazao iznimno uspješan prilikom procesiranja apstraktnih, neurednih i kompleksnih setova podataka.

Dobra računala i procesori važna su komponenta za brzo i efikasno manipuliranje i procesiranje velikih količina podataka koji nisu rijedak slučaj u molekularnoj biologiji. Međutim, ponekad je isplativije uložiti sredstva u kupnju jakih grafičkih kartica. Grafičke kartice mogu izvršavati više matematičkih operacija istovremeno, što omogućava upotrebu paralelizma, odnosno obrade veće količine podataka istovremeno. Paralelizam će drastično povećati efikasnost i procesivnost algoritama kao što su *K*-najbliži susjed, random forests ili neuronske mreže. Međutim, iterativni algoritmi koji svoj rad temelje na nekoliko uzastopnih koraka tu neće profitirati.

Kako bi određeni biološki problem bio uspješno savladan važno je surađivati

sa znanstvenicima i profesijama drugih područja. Programeri bolje poznaju dostupne alate i metode, te će brže i uspješnije prilagoditi algoritam određenoj namjeni, na primjer paralelizaciji. Matematičari i računalni znanstvenici imaju drugačiju percepciju bioloških problema od biologa, a njihovim zajedničkim djelovanjem lakše će se pronaći optimalno rješenje.

6. Literatura

Baker, M., 2010. Next-generation sequencing: adjusting to data overload. Nature methods, 7(7), 495–499.

Breiman L. 2001. Random Forrests, U: Machine learning, Kluwer Academic Publishers, Netherlands 45:5-32.

Cleophas T., Zwinderman A. H. 2014. Machine Learning in Medicine - a Complete Overview, Springer International Publishing, Switzerland.

Frandsen P. B., et al. 2015. Automatic selection of partitioning schemes for phylogenetic analyses using iterative k-means clustering of site rates, BMC Evolutionary Biology, 15:13

Golub T. R., Slonim D. K., et al. 1999. Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring, Science 286, 531-7

Herbrich R., Graepel T. 2012. A First Course in Machine Learning, Taylor & Francis Group, LLC

James G., Witten D., Hastie T., Tibshirani R. 2013. An Introduction to Statistical Learning - with Applications in R, Springer Science+Business Media, New York.

Mukherjee, S., 2003. Classifying Microarray Data Using Support Vector Machines, U: Berrar D. P., Dubitzky W., Granzow M. A Practical Approach to Microarray Data Analysis, Kluwer Academic Publishers, Dordrecht, 166-186.

Naqa I. E., Li R., Murphy M. J. 2015. Machine Learning in Radiation Oncology - Theory and Applications, Springer International Publishing, Switzerland.

Naul B.,2009. A Review of Support Vector Machines in Computational Biology, 1–17. Dostupno na: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.181.201&rep=rep1&type=pdf.

Polak P., Karlić R., Koren A., et al. 2015. Cell-of-origin chromatin organization shapes the Mutational Landscape of Cancer, Nature 518, 360-4.

Rifkin R. Mukherjee S., Tamayo P. et al. 2003. An Analytical Method For Multi-Class Molecular Cancer Classification, SIAM Review 45(4), 706-23.

Šikić M., Tomić S., Vlahoviček K. 2009. Prediction of Protein-Protein Interaction Sites in Sequences and 3D Structures by Random Forests, PLoS computational biology 5(1), e1000278.

Zhang Y-Q., Rajapakse J. C. 2009. Machine Learning in Bioinformatics, John Wiley & Sons, Inc., Hoboken, New Jersey.

7. Sažetak

Strojno učenje je područje računalnih znanosti koje kroz razvoj algoritama omogućava računalima da nauče, modeliraju i razumiju kompleksne skupove podataka. Razni algoritmi strojnog učenja "uče se" izvršavati određeni zadatak na podacima za trening, a naučeno se primjenjuje na novo unesenim podacima. Strojevi programirani algoritmom postaju sve bolji u obavljanju određenih zadataka što imaju više iskustva.

Postoje dvije temeljne podjele algoritama za strojno učenje. Algoritmi s nadzorom pokušat će povezati dva tipa podataka: prediktore i odgovor, odnosno predvidjeti odgovor za novo unesene prediktore. Algoritmi bez nadzora kao konačan cilj nastoji opisati i grupirati unesene podatake. Druga podjela algoritama je na regresijske i klasifikacijske. Algoritmi kojima se razmatraju kvantitativne varijable nazivaju se regresijski algoritmi, a oni koji proučavaju kvalitativne varijable nazivaju se klasifikacijski.

U radu su opisane tri metode strojnog učenja, te su za svaku metodu predstavljeni neki primjeri primjene u molekularnoj biologiji. Random forests prva je od predstavljenih metoda s nadzorom koja se temelji na izgradnji šume stabala odluke. Konačan predviđeni odgovor dobiva se usrednjavanjem odgovora svih stabala. Metoda potpornih vektora za svoj rad koristi kernele, a cilj joj je pronaći optimalnu razdvajajuću hiperravninu i tako što točnije podijeliti opservacije na dvije ili više podgrupa. Poslijednji je algoritam *K*-srednjih vrijednosti koji ća unesene podatke pokušati podijeliti u unaprijed određeni broj podgrupa na temelju međusobne sličnosti.

U zaključku nastojim opisati širu sliku područja strojnog učenja, pružiti neke dodatne zanimljive informacije, ideje i metode, te korisne savjete. Također, ističem važnost pojedinih koraka koji prethode samoj primjeni odabrane metode.

8. Summary

Machine learning is a field of computer science which enables computers to learn how to manipulate and understand complex datasets through algorithm development. Various types of machine learning algorithms are trained on training datasets to apply specific tasks on new input data. Programmed machines gain experience with time which makes them more reliable and successful.

There are two major classifications of machine learning algorithms. Supervised algorithms are trying to relate two types of data: predictors and response. They will estimate the response based on input predictor. On the other hand, unsupervised algorithms are used to group and describe the given dataset.

Second classification is considering regression and classification algorithms. Algorithms that are used for computations with quantitative variables are called regression algorithms, and those manipulating qualitative variables are classification algorithms.

In this bachelor's thesis three machine learning methods are described. For each method examples are given with the application in molecular biology. Random forests is the first supervised method presented. It is based on constructing a forest of decision trees. Final response is estimated by averaging the response of all the trees. Support vector machine is using kernel functions to find the maximum margin hyperplane in order to divide the given dataset into two distinct groups. Last, *K*-means clustering algorithm will tend to divide the given dataset into predefined *K* number of groups based on similarity of the data.

In the conclusion an overview of the machine learning field is given along with the additional information, ideas, methods and advices. Moreover, the importance of the steps that precede the algorithm implementation are highlighted and explained.